**MINISTERUL EDUCAŢIEI AL REPUBLICII MOLDOVA  
Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare Informatică şi Microelectronică  
Catedra Automatica și Tehnologii Informaționale**

RAPORT

**Lucrarea de laborator nr. 4**

**Disciplina: Matematica Discretă  
Tema: Parcurgerea grafului în adâncime și în lărgime.**

**A efectuat: st. gr.AI-151 Brăduleac Vadim**

**A verificat: Dohotaru L.**

**Scopul lucrării:**

1. Studierea metodelor de parcurgere a unui graf: adâncime și lărgime.
2. Elaborarea unei proceduri de introducere a grafului în matricea de adiacență și obținerea parcurgerii acestui graf în adâncime și în lărgime.

**Sarcina de bază:**

1. De elaborat procedura de introducere a unui graf în memoria calculatorului în forma matricei de adiacență.
2. De elaborat procedura de parcurgere a grafului introdus în adâncime și în lărgime
3. Folosind procedurile enumerate ,de elaborat programul care va permite:

>> introducerea grafului reprezentat sub forma liste de adiacență.

>> parcurgerea grafului reprezentat în adâncime și în lărgime.

**Screenshot-uri și listingul aplicației**

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#define MAX 30000

//===================================================================

/\* Declararea structurilor si variabilelor globale \*/

struct List{

int v;

int w;

struct List \*next;

};

struct Graph{

int h;

int p;

struct List \*first;

struct List \*last;

}\*G;

int N,V;

//-------------------------------------------------------------------

/\* Initializarea functiilor \*/

void adincime();

void latime();

void Menu();

void Prezt();

void ListAd();

void ElebList();

//-------------------------------------------------------------------

/\* Corpul principal \*/

int main()

{ Prezt();

Menu();

}

//-------------------------------------------------------------------

/\* Meniul principal \*/

void Menu()

{ int m;

system("color 9");

printf("\n\n");

printf(" |-------------------------------|\n");

printf(" | [1] - Introducerea arborelui.|\n");

printf(" | [2] - Parcurgerea adincime. |\n");

printf(" | [3] - Parcurgerea in latime. |\n");

printf(" | Esc - Iesirea. |\n");

printf(" |-------------------------------|\n");

printf("\n\n");

do m=getch();

while((m<49 || m>51) && m!=27);

switch(m) {

case 49: ListAd(); break;

case 50: adincime(); getch(); break;

case 51: latime(); getch(); break;

return ;

}

Menu();

}

//-------------------------------------------------------------------

/\* Prezentarea temei \*/

void Prezt()

{

printf("\n\n\n\n\t\t");

printf("Lucrare de laborator Nr4 la Matematica Discreta.");

printf("\r\n\n\t\t");

printf("Parcurgerea in adincime si latime.");

getch();

}

//-------------------------------------------------------------------

/\*Parcurgeamea in adincime\*/

void adincime(){

int r ;

system("cls");

printf("Din care nod doriti sa incepeti parcurgerea ?: ");

scanf("%d",&r);

//-------------------------------------------------------------------

/\*Parcurgeamea in latime\*/

void latime(){

int r ;

system("cls");

printf("Din care nod doriti sa incepeti parcurgerea ?: ");

scanf("%d",&r);

//-------------------------------------------------------------------

/\* Lista de adiacenta \*/

void ListAd()

{ int i,v,w;

struct List \*c;

//+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

/\* LstAd \*/

if(G) ElebList();

printf("\n\t");

printf("Dati nr de virfuri a grafului : ");

scanf("%d",&N);

G=(struct Graph \*)malloc(N\*sizeof(struct Graph));

printf("\n\t");

printf("Introduceti lista de adiacenta: \n\n\r");

for(i=0;i<N;i++)

{ printf("%d|",i+1);

G[i].first=(struct List\*)malloc(sizeof(struct List));

G[i].last=G[i].first;

G[i].last->next=NULL;

G[i].last->v=-1;

scanf("%d",&v);;

if(N<v || v<0)

{

printf("\n\n\n\n\n\t\t\t");

printf("Eroare!");

getch();

Menu();

}

while(v)

{ G[i].last->v=v-1;

G[i].last->next=(struct List\*)malloc(sizeof(struct List));

G[i].last=G[i].last->next;

G[i].last->next=NULL;

G[i].last->v=-1;

scanf("%d",&v);

if(N<v || v<0)

{

printf("\n\n\n\n\n\t\t\t");

printf("Eroare!");

getch();

Menu();

}

}

}}

//-------------------------------------------------------------------

/\* Elaborarea listei \*/

void ElebList()

{ struct List \*c,\*t;

while(N--)

{ c=G[N].first;

while(c!=G[N].last)

{ t=c->next;

free(c);

c=t;

}

}

free(G);}

Rezultatele afisarii:

Parcurgerea in adincime.

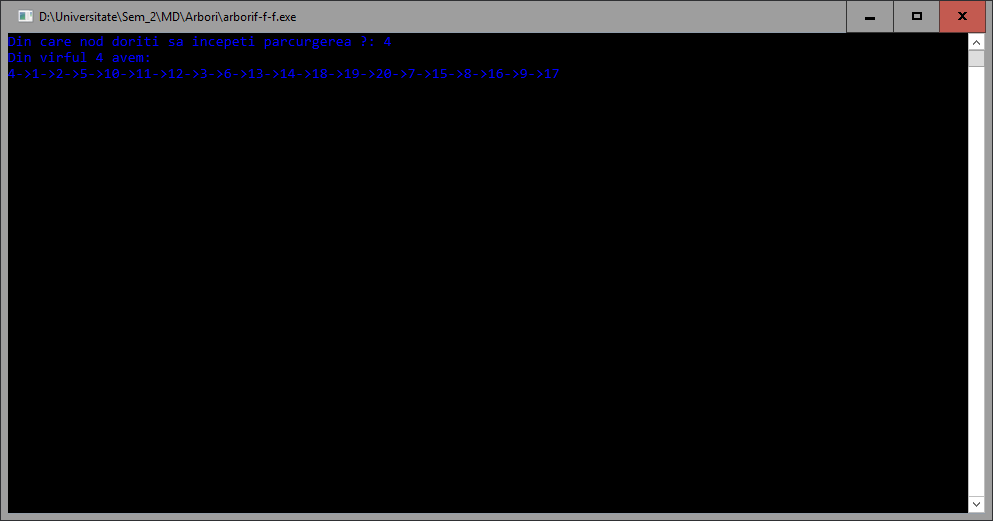


Fig.1

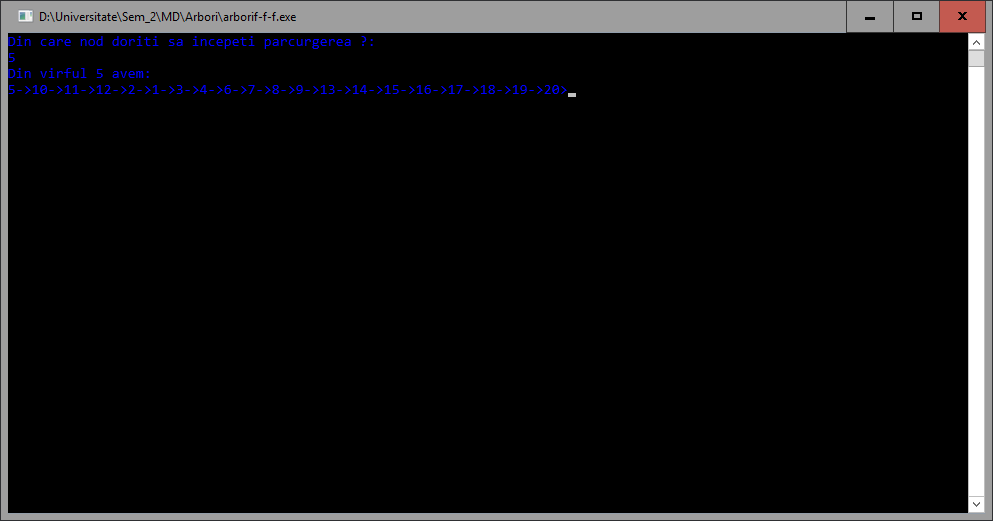
Parcurgerea in latime.  


Fig.2

**Concluzie:**

În urma efectuării acestor lucrări de laborator am învățat să parcurg în adâncime și în lărgime graful(arborele) introdus în memoria calculatorului. De asemenea am căpătat o nouă experiență după realizarea procedurii de parcurgere a unui arbore .